

**Amendments to the Claims**

Please amend Claims 1, 5, 8, 12, 15, 19, and 22-24. The Claim Listing below will replace all prior versions of the claims in the application:

**Claim Listing**

1. (Currently amended) A lookup table comprising:
  - a plurality of mappers which are indexed by successive portions of a search key and partial indexes from prior mappers to output a route index for the search key or to output partial indexes to subsequent mappers; and
  - a partial index feedback loop by which a ~~mapper~~ series of mappers is indexed in multiple passes with multiple successive portions of the search key.
2. (Original) The lookup table as claimed in Claim 1 wherein the route index corresponding to the search key is stored in a single location in one of the plurality of mappers.
3. (Original) The lookup table as claimed in Claim 1 wherein the length of the search key is variable.
4. (Original) The lookup table as claimed in Claim 3 wherein the search key includes a 32-bit IPv4 address.
5. (Currently amended) The lookup table as claimed in ~~Claim 4~~ Claim 3 wherein the route index corresponding to the search key is found after a first search of the plurality of mappers if the length of the search key is less than or equal to the length of a mapper key, and is found after multiple searches of the plurality of mappers if the length of the search key is greater than the length of the mapper key.
6. (Original) The lookup table as claimed in Claim 3 wherein the search key includes a 128-bit IPv6 address.

7. (Original) The lookup table as claimed in Claim 1 wherein the partial index is a subtree index.
8. (Currently amended) A method for providing a longest prefix match for a search key comprising the steps of:
  - providing plural successive portions of the search key to successive mappers with partial indexes from prior mappers to index entries in the ~~mapper~~ successive mappers, each entry storing a route index or a partial index for a subsequent mapper; and
  - feeding back a partial index from a subsequent mapper to a prior mapper to loop back through ~~plural indexes to the prior mapper~~ a series of mappers with plural successive portions of the search key.
9. (Original) The method as claimed in Claim 8 further comprising the step of:
  - outputting the route index corresponding to the search key stored in a single entry in one of the plurality of mappers.
10. (Original) The method as claimed in Claim 8 wherein the length of the search key is variable.
11. (Original) The method as claimed in Claim 10 wherein the search key includes a 32-bit IPv4 address.
12. (Currently amended) The method as claimed in ~~Claim 11~~ Claim 10 wherein the route index corresponding to the search key is output after a first search of the ~~plurality of~~ successive mappers if the length of the search key is less than or equal to the length of a mapper key, and is found after multiple searches of the successive mappers if the length of the search key is greater than the length of the mapper key.

13. (Original) The method as claimed in Claim 10 wherein the search key includes a 128-bit IPv6 address.
14. (Original) The method as claimed in Claim 8 wherein the partial index is a subtree index.
15. (Currently amended) A lookup table comprising:
  - a plurality of mappers which are indexed by successive portions of a search key and partial indexes from prior mappers to output a route index corresponding to the search key or to output partial indexes to subsequent mappers; and
  - means for feeding back a partial index from a subsequent mapper to a prior mapper to loop back through ~~plural indexes to the prior mapper~~ a series of mappers with plural successive portions of the search key.
16. (Original) The lookup table as claimed in Claim 15 wherein the route index corresponding to the search key is stored in a single location in one of the plurality of mappers.
17. (Original) The lookup table as claimed in Claim 15 wherein the length of the search key is variable.
18. (Original) The lookup table as claimed in Claim 17 wherein the search key includes a 32-bit IPv4 address.
19. (Currently amended) The lookup table as claimed in ~~Claim 18~~ Claim 17 wherein the route index corresponding to the search key is found after a first search of the plurality of mappers if the length of the search key is less than or equal to the length of a mapper key, and is found after multiple searches of the plurality of mappers if the length of the search key is greater than the length of the mapper key.

20. (Original) The lookup table as claimed in Claim 17 wherein the search key includes a 128-bit IPv6 address.
21. (Original) The lookup table as claimed in Claim 15 wherein the partial index is a subtree index.
22. (Currently amended) A lookup table providing a route index from a search key comprising:
  - a first mapper which receives a portion of the search key to index an entry which stores a route index corresponding to the search key or stores a first partial index to a next mapper;
  - at least one next mapper which receives a successive portion of the search key and a partial index from prior mappers to index ~~a next mapper~~ an entry of the at least one next mapper which stores the route index corresponding to the search key or stores a next partial index to ~~[[a]]~~ another next mapper; and
  - a selector which selects the next partial index fed back from ~~a next mapper~~ one of the at least one next mappers or the first partial index from the first mapper as the partial index to the at least one next mapper.
23. (Currently amended) An apparatus for providing a route index corresponding to a search key comprising:
  - a forwarding engine which receives the search key and provides a portion of the search key as a mapper key; and
  - a lookup table coupled to the forwarding engine, which receives the mapper key from the forwarding engine, the lookup table comprising:
    - a plurality of mappers which are indexed by successive portions of a search key and partial indexes from prior mappers to output the route index to the forwarding engine for the search key or to output partial indexes to subsequent mappers; and

a partial index feedback loop by which a ~~mapper~~ series of mappers is indexed in multiple passes with multiple successive portions of the search key.

24. (Currently amended) The lookup table of claim 1, wherein ~~the~~ a mapper includes a subtree memory and a subtree mapper, the subtree mapper storing the partial indexes.
25. (Previously presented) The lookup table of claim 1, wherein the partial index includes a pointer to a subtree entry stored in another mapper.